

WHAT ATTRIBUTES INFLUENCE THE SELECTION OF A ROMANTIC
PARTNER?

PREDICT 422 – Section 56 – Group 2
Bruckner, Funk, Sheets, Zimmerman

DATA

- Sourced from [Kaggle](#)
- Compiled by Columbia Business School professors Ray Fisman and Sheena Iyengar for their [paper](#) title “Gender Differences in Mate Selection: Evidence From a Speed Dating Experiment.”

ADDITIONAL DATA INFO

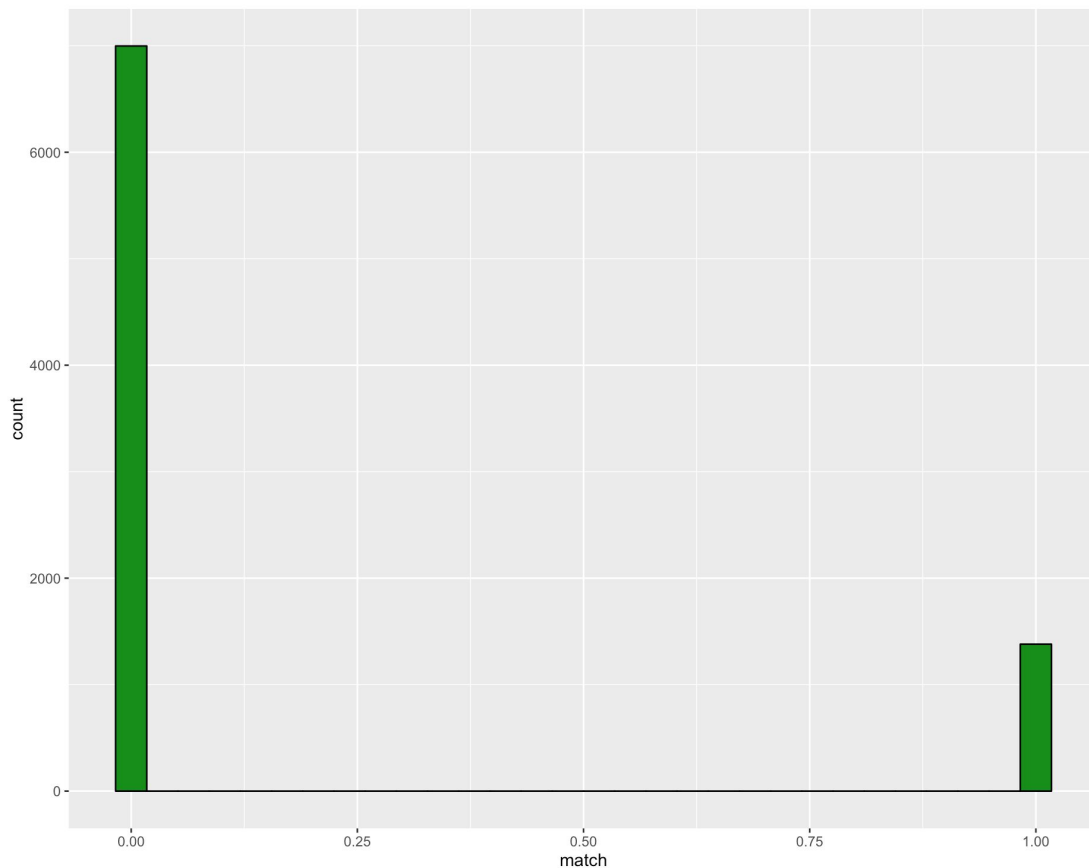
- Data was gathered from participants in experimental speed dating events from 2002-2004. During the events, the attendees would have a four minute "first date" with every other participant of the opposite sex. At the end of their four minutes, participants were asked if they would like to see their date again. They were also asked to rate their date on six attributes: Attractiveness, Sincerity, Intelligence, Fun, Ambition, and Shared Interests.
- The dataset also includes questionnaire data gathered from participants at different points in the process. These fields include: demographics, dating habits, self-perception across key attributes, beliefs on what others find valuable in a mate, and lifestyle information. See the Speed Dating Data Key document below for details.

EDA - INITIAL FINDINGS

- There was not a single line of data that was “complete”.
 - Lots of missing data!
 - Noticed that for a majority of the follow up questions, the data was missing at a higher rate.
 - For simplicity’s sake, decided to focus on variables gathered through the pre-date survey.
 - Demographics, interests, attributes they find important.
 - Additionally, limited to only the waves that used the same preference scale (score from 1-100)
 - Performed listwise deletion instead of imputation to handle missing.
- Data went from 8378 rows with 195 variables (sparsely populated) to 6521 rows with 51 variables (complete).

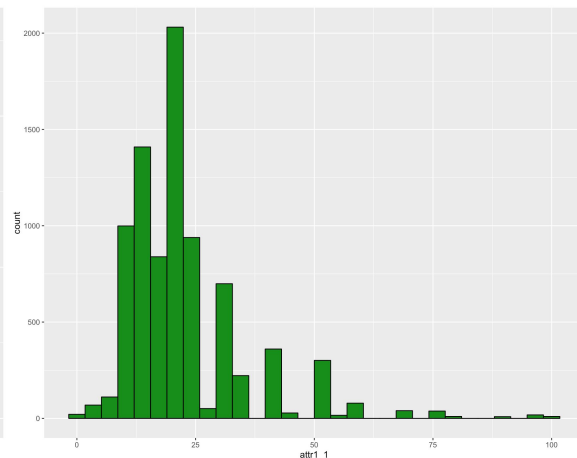
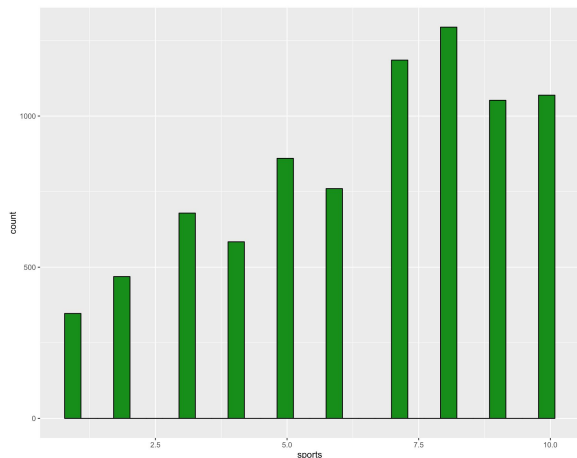
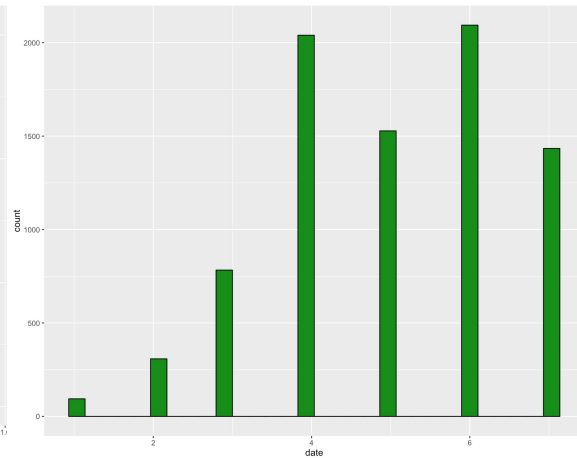
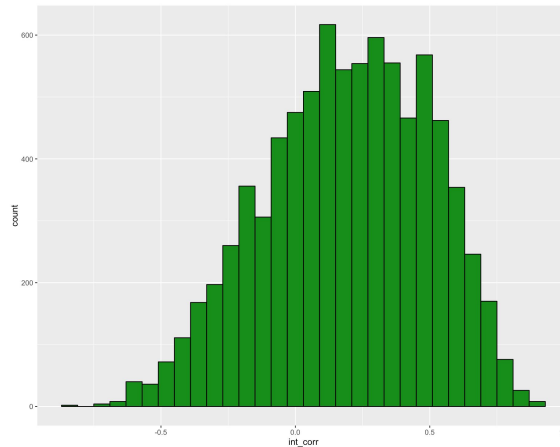
EDA - RESPONSE VARIABLE

- Much fewer matches than non-matches.
- As we'll see in the end results, this greatly impacts our model's ability to successfully predict whether a date will end in a match.
- Classification problem



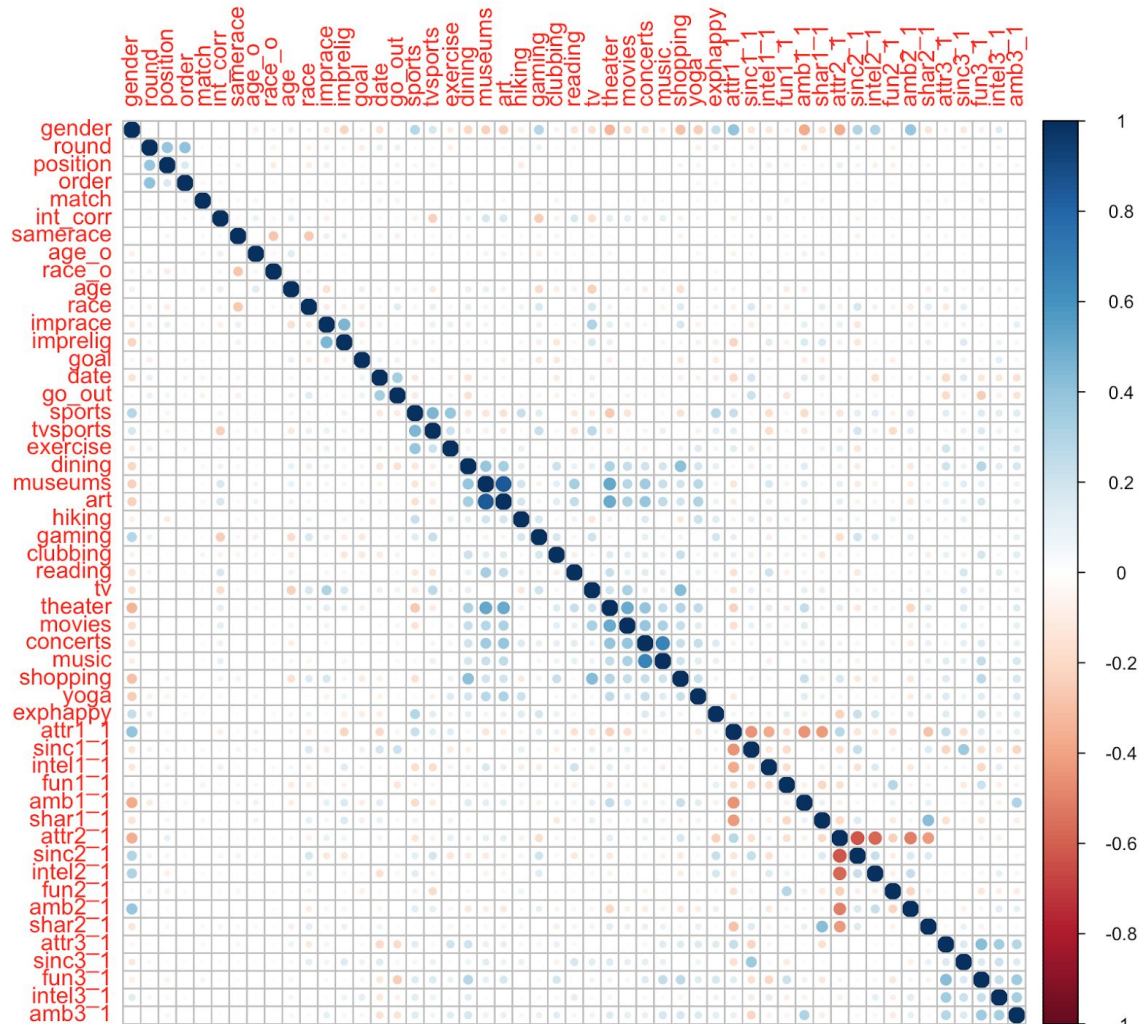
EDA - INPUT VARIABLES

- Demographics: gender, age of participant, age of partner, race of participant, race of partner
- Initial Correlation between interests
- How frequently the participants goes out, on dates & generally.
- What participant is looking for: Date, Meet people, etc.
- Interests: For example, sports
- What the participants looks for in the opposite sex.
- What the participants thinks the opposite sex looks for.
- What participant thinks of themselves.



ANALYSIS - CORRELATIONS

- No variables are even moderately correlated with the response variable, “match”.
- However, some variables are moderately correlated with one another:
 - For example, if someone ranks art as an interest, they are likely to also be interested in museums and theater. Same goes for music and concerts, and theater and movies.
 - Additionally, if someone thinks the opposite sex finds sincerity important, they are likely to believe they do not find attractiveness important in their partner.



MODELING - PREPARATION

- We are trying to predict whether a date will result in a “match” -- an outcome of 1 or 0.
- Split data into train and test data sets.
 - Train is approximately 75% of the data, test is the remaining 25%.
- Center and Scale data to have mean of 0 and stddev of 1
- Classification Techniques Attempted:
 - Logistic Regression using stepwise variable selection
 - K Nearest Neighbor Classification
 - Random Forest
 - Support Vector Machine

MODELING: STEPWISE LOGISTIC

- A majority of the variables are removed during the stepwise variable selection process.
- Additionally, observing the variance inflation factors of the remaining variables in the stepwise logistic model, all of the attributes representing what the participants believe the opposite sex thinks is important have values well above 10 and will be removed from the final model.

```
> vif(step.glm)
```

gender	order	int_corr	samerace	age_o	age	imprace	date	go_out	dining	museums
2.140385	1.011482	1.090534	1.040206	1.045940	1.162631	1.105335	1.292842	1.227202	1.384723	4.389361
art	clubbing	reading	movies	concerts	attr1_1	sinc1_1	shar1_1	attr2_1	sinc2_1	intel2_1
4.097239	1.124215	1.235381	1.371002	1.424144	2.502898	1.569047	1.595997	84.721625	17.696489	15.015277
fun2_1	amb2_1	shar2_1	attr3_1	amb3_1						
14.962845	14.890252	12.787696	1.310381	1.231207						

MODELING: STEPWISE LOGISTIC

- Using a threshold of 0.35, an optimal accuracy can be achieved at 83.9%.
- The sensitivity is AWFUL for this model.
 - It basically cannot seem to find any separating features to distinguish a match.

```
> confusionMatrix(glm.pred,test.std$match,positive='1')
```

Confusion Matrix and Statistics

	Reference	
Prediction	0	1
0	1362	254
1	9	6

Accuracy : 0.8387

95% CI : (0.82, 0.8563)

No Information Rate : 0.8406

P-Value [Acc > NIR] : 0.5964

Kappa : 0.0267

Mcnemar's Test P-Value : <2e-16

Sensitivity : 0.023077

Specificity : 0.993435

Pos Pred Value : 0.400000

Neg Pred Value : 0.842822

Prevalence : 0.159411

Detection Rate : 0.003679

Detection Prevalence : 0.009197

Balanced Accuracy : 0.508256

'Positive' Class : 1

Coefficients:

	Estimate	Std. Error	z value	Pr(> z)	
(Intercept)	-1.68542	0.04077	-41.341	< 2e-16	***
gender	0.05605	0.04536	1.236	0.216519	
order	-0.08676	0.03980	-2.180	0.029280	*
int_corr	0.11081	0.04065	2.726	0.006411	**
samerace	0.10619	0.03912	2.714	0.006643	**
age_o	-0.11250	0.04118	-2.732	0.006298	**
age	-0.16609	0.04384	-3.789	0.000151	***
imprace	-0.13133	0.04148	-3.166	0.001545	**
date	-0.10904	0.04289	-2.542	0.011020	*
go_out	-0.09475	0.04635	-2.044	0.040936	*
dining	0.10886	0.04764	2.285	0.022303	*
museums	-0.19399	0.08171	-2.374	0.017589	*
art	0.17841	0.07932	2.249	0.024508	*
clubbing	0.10192	0.04159	2.450	0.014268	*
reading	0.07368	0.04415	1.669	0.095131	.
movies	-0.14499	0.04399	-3.296	0.000982	***
concerts	0.10529	0.04705	2.238	0.025246	*
attr1_1	-0.15592	0.05524	-2.822	0.004767	**
sinc1_1	-0.08600	0.04718	-1.823	0.068345	.
shar1_1	-0.13738	0.04478	-3.068	0.002158	**
attr3_1	0.07290	0.04513	1.615	0.106227	
amb3_1	-0.06253	0.04313	-1.450	0.147103	

MODELING: KNN

- This model performs slightly better on the test set when using $k=7$.
- The model does really good at predicting a “no match” scenario, but is still struggling capturing the “match” scenario.
- 83.9% accuracy is pretty good.

```
> confusionMatrix(knn.pred,test.std$match,positive='1')
```

Confusion Matrix and Statistics

	Reference	
Prediction	0	1
0	1325	216
1	46	44

Accuracy : 0.8394

95% CI : (0.8206, 0.8569)

No Information Rate : 0.8406

P-Value [Acc > NIR] : 0.5701

Kappa : 0.1846

Mcnemar's Test P-Value : <2e-16

Sensitivity : 0.16923

Specificity : 0.96645

Pos Pred Value : 0.48889

Neg Pred Value : 0.85983

Prevalence : 0.15941

Detection Rate : 0.02698

Detection Prevalence : 0.05518

Balanced Accuracy : 0.56784

'Positive' Class : 1

MODELING: RANDOM FOREST

- This model performs slightly worse than the KNN approach at 82.9% test accuracy.
- Although accuracy is less than the logistic, it does a better job of predicting a match.

```
> confusionMatrix(pred.RF1,test.std$match,positive='1')
```

Confusion Matrix and Statistics

	Reference	
Prediction	0	1
0	1315	223
1	56	37

Accuracy : 0.8289

95% CI : (0.8098, 0.8469)

No Information Rate : 0.8406

P-Value [Acc > NIR] : 0.9055

Kappa : 0.1372

McNemar's Test P-Value : <2e-16

Sensitivity : 0.14231

Specificity : 0.95915

Pos Pred Value : 0.39785

Neg Pred Value : 0.85501

Prevalence : 0.15941

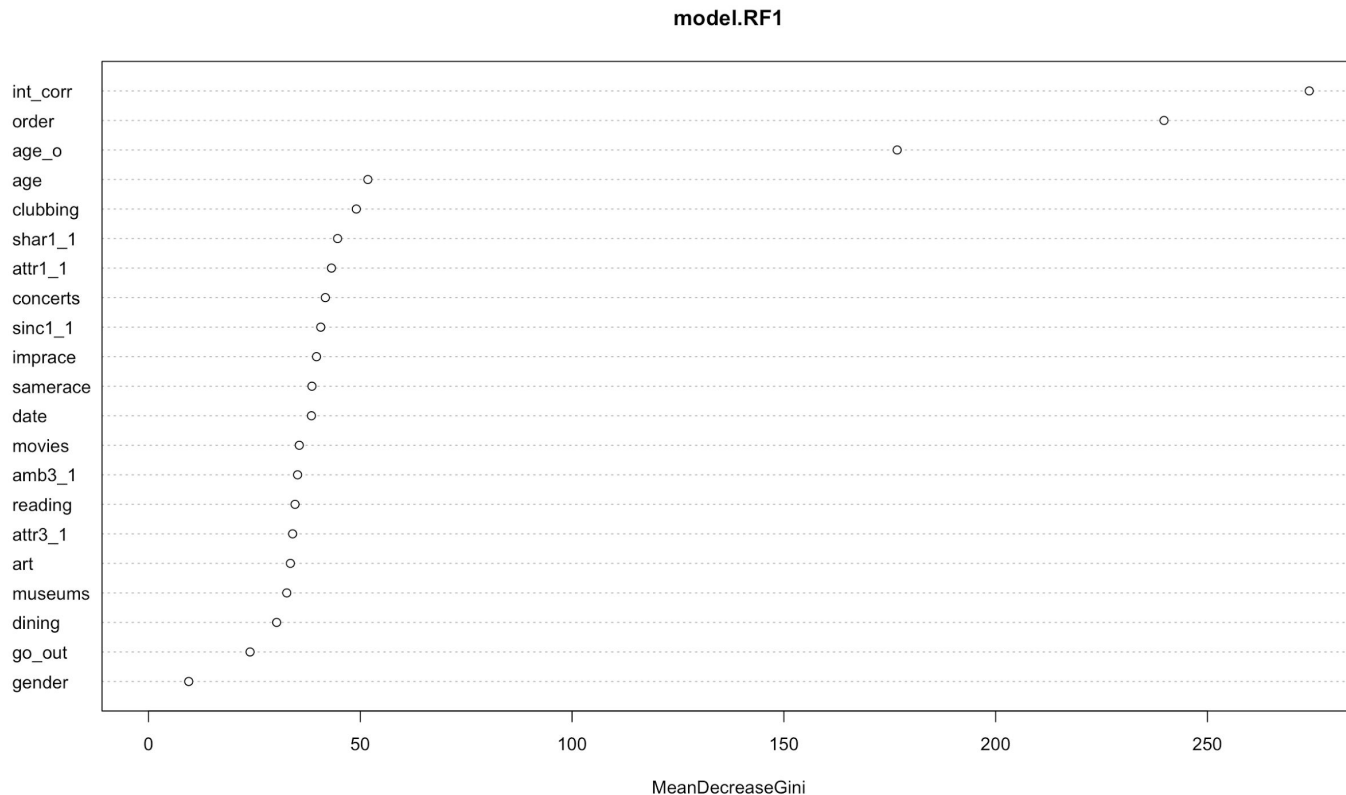
Detection Rate : 0.02269

Detection Prevalence : 0.05702

Balanced Accuracy : 0.55073

'Positive' Class : 1

MODELING: RANDOM FOREST VARIABLE IMPORTANCE



MODELING: SVM TUNING

```
> svm.tune=tune(svm,match~.,data=train.std ,kernel ="radial",ranges =list(cost=c(0.001 , 0.01, 0.1, 1,5,10,100) ))  
> summary(svm.tune)
```

Parameter tuning of 'svm':

- sampling method: 10-fold cross validation

- best parameters:

cost
1

- best performance: 0.1487436

- Detailed performance results:

	cost	error	dispersion
1	1e-03	0.1548789	0.01421086
2	1e-02	0.1547604	0.01419352
3	1e-01	0.1536157	0.01400667
4	1e+00	0.1487436	0.01315632
5	5e+00	0.1535722	0.01430075
6	1e+01	0.1611456	0.01625275
7	1e+02	0.2342837	0.02226606

MODELING: SVM

- The support vector machine is comparable in that it has an accuracy of 83.3%. However, it still is not very good at predicting our target class of “match”.

```
> confusionMatrix(pred.svm$flag,test.std$match,positive='1')
```

Confusion Matrix and Statistics

Prediction	Reference	
	0	1
0	1332	235
1	39	25

Accuracy : 0.832

95% CI : (0.813, 0.8498)

No Information Rate : 0.8406

P-Value [Acc > NIR] : 0.8368

Kappa : 0.0975

Mcnemar's Test P-Value : <2e-16

Sensitivity : 0.09615

Specificity : 0.97155

Pos Pred Value : 0.39062

Neg Pred Value : 0.85003

Prevalence : 0.15941

Detection Rate : 0.01533

Detection Prevalence : 0.03924

Balanced Accuracy : 0.53385

'Positive' Class : 1

AND THE BEST MODEL IS...

- Considering performance in terms of accuracy on the test set alone, one would most likely select the KNN classification model.
- However, considering the interpretability of the random forest (i.e having the variable importances to look back on), one might decide to sacrifice a little bit of accuracy for the insight into what lead an observation to be classified the way that it was.

REFLECTION/SUMMARY/CONCLUSION

- The best model for accuracy isn't always the most interpretable one.
- For support vector machines, it is helpful to leverage R's parallel processing capabilities to speed things up.
- When the data is so heavily skewed towards one class, it throws a wrench into an otherwise classic classification problem.
- Although Gender was speculated as being a variable that impacts whether or not someone gets a match or not, it isn't very important in the random forest and it's not significant at all in the logistic regression.
- Doing this again, we would work on grouping the interests into components representing the category of interest.

CODE

- Full project code can be found here:
https://github.com/amsheets/PREDICT422_GroupECProject